# Artificial Neural Network and Deep Learning

## Homework 3 (A.A. 2020/2021) – Visual Question Answering

Diego Savoia, Matr. 944508 (10535515) – Francesco Emanuele Stradi, Matr. 944616 (10538326)

For this Visual Question Answering problem, the first issue we encountered was the data preparation part. We had to deal with two kind of different inputs: an image and a text for each instance. We implemented a single Custom Dataset to handle both of them plus the output. As concerns the image preparation, we carried on the same steps used in the last challenge; differently, we had to perform tokenization and padding on the input texts. Finally, the output has been converted into integers (one for each answer) to treat the problem as a standard classification task.

As written on Kaggle, the questions/answers can be divided into three categories: yes/no, counting and other. In order to avoid overfitting, it is fundamental that this proportion between classes is maintained in both the training and validation sets. Therefore, in our code you can find two different methodologies for data preparation; the first one is the standard one, splitting questions/image IDs/answers without paying attention to their distribution, while in the second one we preserved the proportions between data by hand. Actually, the final results of the models were pretty similar between these two approaches, even if we noticed that in the second case the training convergence takes more steps; so, in the end we decided to use the first way.

As regards the networks, our work focuses on three different models:

- Model 1 (one LSTM)
- Model 2 (three LSTM)
- Model 3 (using VGG-16) – Final Model

## Model 1 (one LSTM)

For these kind of problems, the network must take in input both images and texts. The features related to the images are extracted by a Convolutional Neural Network, while the features for the texts (after the embedding) are obtained by a Recurrent Neural Network (using a LSTM). Then, they are concatenated and fed to a Fully-Connected Neural Network for classification.

We used an image size of 256x256 and a batch size of 32, since bigger values resulted in out-of-memory errors.
At the beginning, we had very poor results on Kaggle with respect to the validation ones, but then we noticed that during the data preparation for the test set we used a different tokenization than the one used for training. Solved this problem:
→ *Kaggle Score using standard data preparation = 0.52479*
→ *Kaggle Score using class-balanced data preparation = 0.52699*

## Model 2 (three LSTM)

To find different dynamics and so to learn a hierarchical representation of the input texts, we decided to add two others LSTM to the previous model structure. The remaining part of the network is exactly the same as the first model; additionally, an image size of 256x256 and a batch size of 32 have been used. Surprisingly, we obtained very similar results on the validation data.

## Model 3 (using VGG-16) – Final Model

For the image feature extraction part, we decided to implement the model using a Transfer Learning approach (fine-tuning). The architecture chosen is VGG-16, which showed good results in the previous challenges.

→ *Kaggle Score with image size of 256x256, batch size of 32 and 512 units in each LSTM = 0.57972*
→ *Kaggle Score with image size of 256x256, batch size of 32 and 256 units in each LSTM = 0.58270*
→ *Worse performance with a learning rate = 1e-4*
→ *Kaggle Score with image size of 256x256, batch size of 32 and 256 units in each LSTM but using the class-balanced data preparation = 0.58364*
→ *Kaggle Score with image size of 400x700, batch size of 16 and 256 units in each LSTM = 0.59384 (best model)*

Where not specified, the standard data preparation procedure has been used.